

NAUCZANIE INFORMATYKI BEZ KOMPUTERA

Michał Piskorski, Martyna Paliwoda, Barbara Kołodziejczak
Koło Naukowe Robotyki
Wydział Matematyki i Informatyki
Uniwersytet im. Adama Mickiewicza w Poznaniu,
ul. Uniwersytetu Poznańskiego 4, 61-614 Poznań
<https://www.facebook.com/KoloNaukoweRobotykiWMI>

Abstract. *The Computer Science Unplugged (CSU) approach is an effective method for fostering computational thinking, problem-solving skills, and algorithmic competencies. Research indicates that activities conducted without the use of computers can achieve learning outcomes comparable to, or even better than, those of traditional computer-based instruction. By employing active learning methods and connecting concepts to students' everyday experiences, CSU serves as a valuable complement to contemporary computer science education.*

1. Wstęp

Współczesna edukacja informatyczna coraz częściej koncentruje się nie tylko na obsłudze urządzeń cyfrowych, ale przede wszystkim na rozwijaniu logicznego myślenia, analizowania problemów oraz rozumienia mechanizmów działania technologii [6]. Jednocześnie praktyka szkolna pokazuje, że wielu uczniów postrzega informatykę, przede wszystkim algorytmikę i programowanie, jako dziedzinę trudną, abstrakcyjną i przeznaczoną wyłącznie dla osób o specjalnych predyspozycjach. W odpowiedzi na te wyzwania powstała idea *Computer Science Unplugged* (CSU), czyli nauczania informatyki bez użycia komputera [1].

Projekt CSU został opracowany w latach dziewięćdziesiątych XX wieku przez Timą Bella, Ianą H. Witteną oraz Mike'a Fellowsa na Uniwersytecie Canterbury w Nowej Zelandii [1]. Jego twórcy zauważyli, że wiele kluczowych zagadnień informatycznych można skutecznie wyjaśniać poprzez doświadczenie, ruch, współpracę i zabawę, bez konieczności korzystania ze sprzętu komputerowego. Co więcej, ograniczenie użycia komputera pozwala uczniom skupić uwagę na samym mechanizmie działania algorytmów oraz procesie rozwiązywania problemów.

Zajęcia realizowane zgodnie z ideą CSU dotyczą m.in. algorytmów sortowania, kodowania i kompresji danych, grafiki komputerowej, kryptografii czy działania sieci Internet [1]. Aktywności te odwołują się do sytuacji bliskich codziennym doświadczeniom

uczniów, dzięki czemu nawet abstrakcyjne zagadnienia informatyczne stają się bardziej zrozumiałe i angażujące. W zależności od wieku uczestników mogą one przyjmować różnorodne formy, takie jak zabawy dydaktyczne, gry edukacyjne, eksperymenty czy edukacyjne escape roomy, wspierające naukę poprzez aktywne działanie i współpracę.

Podejście zgodne z ideą CSU wpisuje się również w założenia obowiązującej podstawy programowej [5], która podkreśla znaczenie rozwijania myślenia algorytmicznego, analizowania problemów oraz tworzenia strategii ich rozwiązywania. Nauczanie informatyki bez komputera może stanowić wartościowy etap przygotowujący uczniów do późniejszej nauki programowania oraz pracy z bardziej zaawansowanymi narzędziami cyfrowymi.

2. Metoda CSU a podstawa programowa

W polskiej podstawie programowej ujęto szereg celów kształcenia, do których realizacji nie jest wymagane użycie komputera [5]. Na etapie klas IV–VIII od uczniów oczekuje się m.in. rozwiązywania problemów w oparciu o myślenie logiczne, formułowania codziennych czynności w postaci algorytmów, a także praktycznego stosowania i porównywania metod wyszukiwania oraz porządkowania danych. Ponadto podstawa wymaga znajomości ogólnej architektury i usług sieci Internet [5]. Co istotne, dokument ten definiuje jedynie efekty kształcenia nie narzucając nauczycielom ścisłych metod pracy. Stanowi to doskonałą przestrzeń do wdrażania założeń edukacji bez komputera (CSU) [1]. W niniejszym artykule zaprezentowano scenariusze lekcji z wybranych zagadnień na poziomie szkoły podstawowej i ponadpodstawowej.

3. Jak waga może pomóc w sortowaniu?

Jak zauważa Maciej M. Sysło w swojej książce *Algorytmy* [7], operacje porządkowania danych mają wpływ na sprawność funkcjonowania wielu obszarów naszej działalności. Uporządkowanie zbioru – co na co dzień możemy zaobserwować w dziennikach lekcyjnych czy katalogach bibliotecznych – znacząco przyspiesza operacje takie, jak wyszukiwanie, dodawanie czy usuwanie elementów. W informatyce problem ten można rozwiązać za pomocą algorytmów o różnej efektywności. Doskonałym sposobem na pokazanie uczniom istoty optymalizacji jest analiza algorytmu Quicksort [7, 10].

3.1. Szybki algorytm sortowania

Fundamentem algorytmu sortowania szybkiego (ang. Quicksort) jest jedna z najważniejszych strategii w informatyce – metoda „dziel i zwyciężaj” [7]. Algorytm ten opiera się na inteligentnym podziale problemu na mniejsze, łatwiejsze do przetworzenia części. Proces rozpoczyna się od wyboru ze zbioru elementu odniesienia, zwanego elementem dzielącym (ang. pivot). Następnie pozostałe wartości są z nim porównywane: elementy mniejsze trafiają na lewą stronę, a większe (lub równe) na prawą. W efekcie

tego pojedynczego podziału, element dzielący natychmiast zajmuje swoje ostateczne, docelowe miejsce w ciągu, a główny problem redukuje się do dwóch niezależnych podgrup. Dla każdej z nich cały proces jest powtarzany rekurencyjnie [7].

Aby przełożyć niełatwą koncepcję algorytmu na język zrozumiały dla uczniów (np. klasy siódmej szkoły podstawowej) i przygotować ich do działań praktycznych bez użycia komputera, można zapis algorytmu sprowadzić do czterech klarownych kroków [1]:

- Wybór elementu dzielącego: Wybierz pierwszy element w rozpatrywanym ciągu jako punkt odniesienia (jest to tzw. „Kapitan”).
- Podział grupy: Porównaj z nim każdy kolejny element w zbiorze. Wartości mniejsze odłóż na lewą stronę, a większe na prawą.
- Miejsce docelowe: Umieść element dzielący dokładnie pomiędzy lewą a prawą grupą. Od tego momentu jego pozycja jest posortowana i niezmienna.
- Rekurencja: Traktuj utworzoną lewą i prawą podgrupę jako nowe, zupełnie odzielne zadanie. Zastosuj wobec nich kroki od 1 do 3 i powtarzaj ten proces tak długo, aż w wydzielonych grupach pozostanie najwyżej jeden element [1].

Dzięki takiemu podejściu największa liczba operacji wykonywana jest na systematycznie pomniejszanych fragmentach zbioru, co radykalnie skraca całkowity czas układania danych. Analiza tego algorytmu uświadamia uczniom znaczenie optymalizacji oraz pokazuje, jak trafny dobór odpowiedniej strategii logiczno-matematycznej wpływa na ogólną wydajność rozwiązywania problemów [10].

3.2. Scenariusz lekcji

Streszczenie: Systemy informatyczne nieustannie przetwarzają i porządkują ogromne zbiory danych. Wybór nieefektywnej metody sortowania może znacząco obniżyć wydajność systemu, niezależnie od mocy obliczeniowej sprzętu. Podczas proponowanych zajęć uczniowie testują w praktyce działanie algorytmu sortowania szybkiego (Quick-sort) opartego na strategii „dziel i zwyciężaj”. Symulacja realizowana jest bez użycia komputera – uczestnicy wykorzystują tradycyjną wagę szalkową oraz karty pracy służące do bieżącego monitorowania liczby wykonywanych operacji.

Grupa docelowa: Uczniowie klas IV–VIII szkoły podstawowej (11 lat i więcej).

Pomoce dydaktyczne: Każdy zespół roboczy otrzymuje:

Zestaw 7 nieprzezroczystych woreczków wypełnionych materiałem sypkim (np. ryżem) w taki sposób, aby każdy z nich miał inną masę.

Tradycyjną wagę szalkową.

Wydrukowaną kartę pracy do rejestrowania liczby operacji porównania (czyli liczby ważeń).

Przebieg lekcji

Wprowadzenie: Nauczyciel prezentuje uczniom zestaw woreczków i pyta o najlepszy sposób ułożenia ich od najlżejszego do najcięższego. Następnie wyjaśnia, że komputer nie analizuje całego zbioru danych jednocześnie, gdyż procesor w pojedynczym cyklu obliczeniowym potrafi porównać ze sobą wyłącznie dwie wartości. Waga szalkowa doskonale modeluje to sprzętowe ograniczenie. Zadaniem uczniów jest posortowanie elementów przy wykonaniu jak najmniejszej liczby ważeń. Każde użycie wagi musi zostać na bieżąco odnotowane w karcie pracy.

Sortowanie w grupach: Uczniowie dzielą się na kilkuosobowe zespoły. Aby zminimalizować liczbę ważeń, stosują algorytm Quicksort, realizując następujące kroki:

Wybór elementu dzielącego: Uczniowie losują z grupy jeden woreczek, który staje się ich punktem odniesienia (w narracji dydaktycznej nazywany „Kapitanem”).

Podział zbioru: Umieszczają wybrany element dzielący na jednej z szalek wagi. Następnie kolejno porównują z nim pozostałe woreczki z badanej grupy. Każde pojedyncze użycie wagi jest natychmiast odnotowywane w karcie pracy poprzez postawienie kreski. Woreczki lżejsze od elementu dzielącego odkładane są na stół po lewej stronie, a cięższe po prawej.

Miejsce docelowe: Po porównaniu wszystkich elementów w grupie, woreczek pełniący funkcję elementu dzielącego umieszczany jest dokładnie pomiędzy zbiorem elementów lżejszych i cięższych. Od tego momentu jego pozycja jest ostateczna (jest on posortowany) i nie ulega już zmianie.

Rekurencja: Zespoły traktują zbiory po lewej i prawej stronie jako nowe, niezależne zadania. Proces wyboru elementu dzielącego i podziału na lżejsze/cięższe powtarzany jest dla każdej z utworzonych podgrup, przy ścisłym zachowaniu zasady ciągłego notowania liczby ważeń.

Zakończenie: Ćwiczenie dobiega końca, gdy w wyniku kolejnych podziałów powstaną wyłącznie zbiory jednoelementowe lub puste (co oznacza pełne posortowanie wszystkich elementów). Nauczyciel podsumowuje zajęcia na podstawie danych zebranych w kartach pracy. Klasa wspólnie analizuje, w jaki sposób systematyczne dzielenie dużego problemu na mniejsze części pozwala znacząco zredukować liczbę niezbędnych operacji.

Komentarz dydaktyczny: Zaproponowana metoda pracy pozwala uczniom przyswoić, nierzadko uważany za trudny, algorytm szybkiego sortowania. Odejście od złożonej analizy teoretycznej na rzecz ukierunkowanej aktywności fizycznej sprawia, że mechanika działania algorytmu staje się dla nich w pełni zrozumiała. Wykorzystana w ćwiczeniu waga szalkowa pełni funkcję procesora, który w danej chwili może porównać jedynie dwa obiekty. Jednocześnie zastosowanie karty pracy kształtuje postawę badawczą uczestników, a precyzyjne notowanie każdego użycia wagi pozwala w praktyce wyliczyć „koszt” działania algorytmu. Pokazuje to w namacalny sposób, że celowa dekompozycja

problemu, charakterystyczna dla strategii „dziel i zwyciężaj”, stanowi optymalną metodę szybkiego porządkowania danych.

KARTA PRACY: SZYBKIE WAŻENIE (QUICKSORT)

Grupa:

Cel: Ułożenie woreczków od najszybszego do najcięższego. Komputer (waga szalkowa) potrafi porównać tylko dwa woreczki naraz! Waszym zadaniem jest zrobić to sprytnie, wykonując jak najmniej ważeń.

Instrukcja obsługi wagi (Kroki Algorytmu):

1. **Wybierz Kapłana:** Wylosuj w ciemno jeden woreczek z grupy. Nie waż go, po prostu pokaż przed wagą jako punkt odniesienia.
2. **Waż i notuj:** Kładź Kapłana na jednej szalce. Na drugą kładź po kolei pozostałe woreczki z tej grupy. **ZA KAŻDYM RAZEM postaw kresek (|) w tabeli poniżej!**
3. **Rozdzielaj:** Woreczki cięższe od Kapłana odkładaj na stół po jego LEWEJ stronie, a ciębsze po PRAWIEJ.
4. **Zamień Kapłana:** Gdy sprawdzisz całą grupę, Kapłan jest już na swoim docelowym miejscu. Od teraz jest "zamieszony" – nie waż go już więcej!
5. **Powtarzaj (Rekurencja):** Potraktuj woreczki po lewej stronie jako zupełnie nową grupę. Wybierz dla nich nowego (Pod)Kapłana i powtórz kroki 1–4. Potem zrób to samo dla prawej strony. Rób tak, aż zostaną pojedyncze woreczki.

Etap sortowania	Zliczanie ważeń (stawiaj kreski)
Krok 1: Główny Kapłan i pierwszy podział na leższe/ciejsze	
Krok 2: Wybór Pod-Kapłanów (dla grupy lewej i prawej)	
Krok 3: Kolejne podziały podgrup	
Krok 4: Ostatnie sprawdzające ważenia (jeśli były potrzebne)	
CAŁKOWITA LICZBA WAŻEŃ (SUMA):	

Wnioski po wykonaniu zadania:

1. Ile wyniosła łączna liczba wszystkich koniecznych użyć wagi?

2. Co działa się z wielkością (liczbą elementów) nieposortowanych grup po każdym prawidłowym wyborze Kapłana?

Rysunek 1. Karta pracy do zliczania liczby operacji

4. DNS – sprawdź kim jestem

W dobie powszechnej cyfryzacji zrozumienie architektury sieci Internet jest fundamentem nowoczesnej edukacji informatycznej. Choć użytkownicy operują na co dzień przyjaznymi, słownymi nazwami stron, urządzenia sieciowe komunikują się ze sobą wyłącznie za pomocą numerycznych adresów IP. Pomostem między nimi jest Domain Name System (DNS, pol. system nazw domen), czyli rozproszona, hierarchiczna baza danych, pełniąca funkcję globalnej „książki telefonicznej”. Wprowadzenie tych pojęć na zajęciach w szkole ma kluczowe znaczenie nie tylko z perspektywy technicznej, ale przede wszystkim w kontekście profilaktyki cyberzagrożeń [4].

4.1. Struktura nazw domenowych

Każdy adres internetowy posiada ściśle określoną budowę. Kluczowym aspektem z punktu widzenia dydaktyki jest uświadomienie uczniom, że system DNS analizuje nazwę domeny w sposób odwrotny, niż czyta to człowiek, a mianowicie od strony prawej do lewej (przechodząc od ogółu do szczegółu) [4].

Analizując przykładowy adres uniwersytecki *amu.edu.pl*, można zaobserwować następującą hierarchię: na samym końcu adresu znajduje się niewidoczna kropka reprezentująca domenę główną (ang. Root Domain). Przed nią widnieje domena najwyższego poziomu (TLD - ang. Top-Level Domain) wskazująca na kraj (.pl), następnie funkcjonalna domena drugiego poziomu przeznaczona dla instytucji edukacyjnych (.edu), a dopiero na początku subdomena określająca konkretną organizację (amu). Zrozumienie tej hierarchii to dla ucznia podstawowa i najskuteczniejsza metoda obrony przed atakami socjotechnicznymi, takimi jak phishing czy typosquatting (rejestrowanie fałszywych domen z celowymi literówkami, np. *fcaebok.com.logowanie.pl*) [4].

4.2. Scenariusz lekcji

Streszczenie: Architektura sieciowa bywa dla uczniów trudna do przyswojenia podczas tradycyjnego wykładu. W niniejszym scenariuszu proces komunikacji między serwerami DNS pokazano w postaci edukacyjnego escape roomu. Uczniowie fizycznie wędrują po sali lekcyjnej między stacjami symbolizującymi kolejne węzły sieci. Weryfikując fałszywe adresy URL, odkrywają drogę od serwera głównego (root server) aż po docelowy adres IP.

Grupa docelowa: Uczniowie szkół ponadpodstawowych.

Pomoce dydaktyczne: Zestawy odpowiednio oznaczonych i ukrytych w sali kopert z interdyscyplinarne projekty STEAM zadaniami dla poszczególnych grup, symbolizujących: serwery główne (root servers), serwery TLD oraz serwery autorytatywne. Skrzynka zamknięta na kłódkę z czterocyfrowym kodem resetowalnym.

Przebieg lekcji

Wprowadzenie: Nauczyciel dzieli klasę na cztery zespoły (np. Czerwony, Niebieski, Zielony, Żółty). Każda z grup otrzymuje zadanie: znalezienie poprawnego adresu IP znanej instytucji, która jest im przypisana (np. Centralnej Komisji Egzaminacyjnej). Nauczyciel wyjaśnia, że sala lekcyjna stała się siecią Internet, a uczniowie pełnią rolę zaopatrników DNS poszukujących właściwego serwera.

Etap 1: Serwer główny (root server): Zespoły otwierają przypisane im koperty startowe. Odnajdują w nich definicję serwera głównego oraz zestawienie kilku ładząco podobnych adresów URL. Zadaniem uczniów jest odrzucenie spreparowanych przez hakerów linków phishingowych poprzez sprawdzenie, co dokładnie znajduje się przed domeną najwyższego poziomu. Odkrycie prawdziwego, bezpiecznego adresu wskazuje im kolejną kopertę (serwer DNS) ukrytą w sali.

Etap 2: Serwer domeny najwyższego poziomu (TLD): Po dotarciu do właściwych kopert (reprezentujących domeny najwyższego poziomu, np. .pl), uczniowie stają przed wyzwaniem związanym z atakiem typu typosquatting. Z puli linków zawierających ce-

lowe błędy i drobne literówki muszą wyłonić oryginalny, poprawny adres docelowy. Rozwiązanie tej zagadki zaowocuje odnalezieniem subdomeny szukanego adresu i skieruje uczniów do kolejnej stacji.

Etap 3: Serwer autorytatywny: W ostatniej kopercie zespoły odnajdują poszukiwany adres IP swojej instytucji. Jedna z cyfr tego adresu jest, na karcie każdego zespołu, wyraźnie wyróżniona i stanowi fragment czterocyfrowego szyfru do kłódki blokującej dostęp do głównej skrzynki.

Finał: Żadna z grup nie posiada pełnego szyfru do kłódki. Aby zakończyć symulację i otworzyć skrzynkę, w której ukryta jest nagroda, uczniowie muszą współpracować, aby złamać szyfr.

Komentarz dydaktyczny: Przekształcenie niełatwej dla uczniów teorii z zakresu hierarchicznej struktury nazw domenowych w edukacyjną grę terenową może prowadzić do dużej skuteczności dydaktycznej. Działanie złożonego systemu DNS przyjmuje w symulacji formę fizycznej drogi do pokonania, co pozwala uczniom lepiej zrozumieć sposób funkcjonowania algorytmu rozwiązywania nazw domenowych. Połączenie wiedzy o domenach z praktycznym wykrywaniem zagrożeń cybernetycznych sprawia, że nauka powiązana jest z rzeczywistością. Dodatkową wartością wychowawczą scenariusza jest wymóg współpracy zespołów na finiszu gry, co promuje umiejętności interpersonalne i ma uświadomić uczniom fakt, że bezpieczeństwo w sieci opiera się na wspólnym działaniu a nie izolacji.

● WIADOMOŚĆ Z SERWERA GŁÓWNEGO (ROOT :.) ●

Miejsce Grupy 1: Zdobądźcie adres IP serwera Centralnej Komisji Egzaminacyjnej.

UWAGA: Skutkanie i otwieranie w sali TYLKO kopercy w kolorze CZERWONYM! Zanim wysuniecie w drogę do kolejnego serwera, musicie udowodnić, że rozpoznajecie zagrożenia w sieci.

POBRANY FRAGMENT WIEDZY: Co robi Serwer Root?

Serwer Główny (Root Server, oznaczany kropką „.”) to najwyższy szczebel w hierarchii DNS. To tutaj zaczyna się każda podróż! Serwer ten nie wie, jak „dobry” adres IP ma Wasza strona, ale posiada kompletną listę serwerów TLD (Top-Level Domain). Gdy zapytacie go o domenę oile.goupl, on „odetnie” końcówkę i skieruje Was do serwera odpowiedzialnego za polskie domeny .pl.

Ściąga Cyber-Detektywa:

- Adres internetowy czytamy zawsze **OD PRAWIEJ DO LEWEJ** (od ogółu do szczegółu).
- Ostatni człon po kropce to Domena Najwyższego Poziomu (np. .pl).
- PRZESTRZEŻNIE!** Prawdziwa domena docelowa musi znajdować się bezpośrednio przed domeną najwyższego poziomu! Oczuliło coś? Dodaj prawdziwą nazwę na początku drugiego, faktycznego adresu.

🚨 Zadanie do wykonania:

Hackerzy próbują wykraść hasła uczniów do e-dziennika Vulcan. Tylko jeden z poniższych adresów logowania jest prawdziwy i bezpieczny. Przeznalcujcie je zgodnie z zasadą „od prawej do lewej” i postawcie znak „X” w nawiasie przy poprawnej odpowiedzi!

() vulcan.edu.pl.logowanie.com	→ otwórzcie czerwony kopertę R
() dziennik-vulcan.edu-pl.org	→ otwórzcie czerwony kopertę W
() uonetplus.vulcan.edu.pl	→ otwórzcie czerwony kopertę X
() logowanie.vulcan.edu.pl.net	→ otwórzcie czerwony kopertę N

🔑 KROK DO PRZODU:

Udajcie się do czerwonej kopercy oznaczonej niebieską strzałką do Waszej odpowiedzi. Uwaga! – błąd zaprowadzi Was na zawiązany serwer i stracie ceny czas!

Rysunek 2. Przykładowa karta startowa dla jednej z grup

5. Kto do pary?

Chociaż *problem trwałego małżeństwa* (również model dwustronnego dopasowywania) nie należy do kanonu algorytmów szkolnych, jednak szereg jego zastosowań w dziedzinach takich, jak ekonomia, matematyka czy informatyka, a także w życiu codziennym, czyni go niezwykle interesującym. Algorytm znajdowania stabilnego dopasowania ma zastosowanie np. przy rekrutacji do szkół, przydziału stażystów do szpitali, a nawet podczas kojarzenia dawców organów z biorcami przeszczepów [9].

5.1. Problem trwałego małżeństwa

Problem trwałego małżeństwa został zdefiniowany w następujący sposób:

Biorąc pod uwagę n mężczyzn i kobiet, gdzie każda osoba uszeregowwała wszystkich członków płci przeciwnej w określonej sekwencji preferencji. Dopasuj w pary małżeństw kobiety i mężczyzn tak, aby nie było dwóch osób przeciwnej płci, które wołałyby mieć siebie nawzajem zamiast obecnych partnerów. Gdy nie ma takich par, zbiór małżeństw uznaje się za stabilny [3].

W 1961 r. David Gale i Lloyd Shapley [3] udowodnili, że problem trwałego małżeństwa ma zawsze rozwiązanie i przedstawili algorytm, który takie rozwiązanie generuje zwany *algorytmem odroczonej akceptacji* (deferred acceptance algorithm). Można go przedstawić w następujących krokach:

Krok 1. Każdy mężczyzna oświadcza się kobiecie, która znajduje się na szczycie w jego rankingu.

Krok 2. Każda kobieta zapisuje otrzymywane propozycje.

Krok 3. Gdy wszyscy mężczyźni złożyli propozycje swoim wybrankom, każda kobieta, która otrzymała więcej niż jedną propozycję uprzejmie odmawia zapisanym mężczyznom oprócz tego, który znajduje się najwyżej w jej rankingu. Nie oznacza to jeszcze jego całkowitej akceptacji, chyba że jest to mężczyzna najbardziej preferowany spośród wszystkich potencjalnych kandydatów.

Krok 4. Odrzuceni zalotnicy składają propozycję następnej kobiecie w ich rankingu.

Krok 5. Przejdź do punktu 2 lub zakończ algorytm, jeżeli każda kobieta wybrała przyszłego męża [9].

Powyższy algorytm odroczonej akceptacji daje zawsze najlepsze możliwe stabilne pary z punktu widzenia mężczyzn. Odwrócenie ról mężczyzn i kobiet doprowadziłoby do powstania stabilnego dopasowania, które byłoby preferowane przez kobiety.

5.2. Scenariusz lekcji

Streszczenie: Z problemem doboru uczniów w pary mamy często do czynienia w praktyce szkolnej. Nie zawsze efekt końcowy zadowala obie strony. Warto więc sięgnąć po sprawdzone metody, które wspomogą nauczycieli w tej niewdzięcznej roli. Celem zajęć

jest zapoznanie uczniów z algorytmem znajdowania trwałych par oraz pokazania jego zastosowań w życiu codziennym.

Grupa docelowa: Uczniowie klas IV–VIII szkoły podstawowej (11 lat i więcej).

Pomoce dydaktyczne: karteczki w dwóch różnych kolorach, długopisy, guziki w czterech kolorach do losowego podziału uczniów na grupy, zasady gry (czyli algorytm znajdowania trwałych par) zapisany językiem zrozumiałym dla uczniów danej grupy wiekowej

Przebieg lekcji

Wprowadzenie: Nauczyciel/ka wprowadza uczniów w temat mówiąc: “Chciałbym/chciałabym żebyście dobrali się w pary, w których będziecie pracować nad projektem. Uczniowie spontanicznie dobierają się w pary. Należy oczekiwać, że część uczniów zostanie bez pary, a inni będą tworzyć bardziej liczne zespoły. Wówczas nauczyciel/ka proponuje grę, która pozwoli utworzyć pary w oparciu o preferencje co do partnerów w projekcie.

Etap 1: Uczniowie zostają podzieleni na dwa równoliczne zespoły A i B w drodze losowania guzików z woreczka. Osoba, która zostanie bez pary będzie czuwać nad przestrzeganiem zasad gry.

Etap 2: Każda osoba otrzymuje kartkę z opisem zasad gry. Uczniowie czytają zasady. Następnie nauczyciel/ka omawia z uczniami kolejne kroki gry, wyjaśnia ewentualne wątpliwości. Podsumowaniem tego etapu może być sformułowanie strategii obu zespołów:

Strategia grupy A: Złóż propozycję osobie z grupy B, która jest najwyżej na Twojej liście preferencji, a która jeszcze nie odrzuciła Twojej propozycji.

Strategia grupy B: Jeśli otrzymałaś/eś więcej niż jedną propozycję od osób z grupy A, to przyjmij ją od osoby, która jest najwyżej na Twojej liście preferencji. Pozostałym osobom grzecznie odmów. Przyjęcie propozycji nie jest ostateczne - możesz jeszcze zmienić decyzję, jeśli otrzymasz propozycję od osoby, która jest wyżej na Twojej liście preferencji.

Etap 3: Uczniowie otrzymują kolorowe karteczki. Każda grupa ma kartki w innych kolorach. Każda osoba (w obu grupach) sporządza listę preferencji osób z grupy przeciwnej, czyli zapisuje w jakiej kolejności poprosiłaby osoby z grupy przeciwnej, aby zostali jej/jego partnerem/partnerką w projekcie.

Etap 4: Uczniowie dobierają się w pary korzystając z listy preferencji i zasad gry. Nauczyciel/ka czuwa nad przestrzeganiem zasad gry. Gra kończy się, gdy wszystkie osoby z grupy A są sparowane z osobami z grupy B.

Podsumowanie gry: Nauczyciel/ka pyta: “Jak sądzicie, czy odwrócenie ról w grze prowadzi do takich samych układów par?” Następnie grupy zamieniają się rolami, czyli

teraz osoby z grupy B wybierają partnerów, a osoby w grupie A zbierają propozycje i dokonują wyboru.

Zakończenie: Nauczyciel/ka pyta: "Jak sądzicie, gdzie Waszym zdaniem można wykorzystać poznany algorytm?" Nauczyciel podaje również kilka własnych przykładów, jak: wybór partnerów do tańca towarzyskiego, wybór szkoły średniej, wybór współlokatora na szkolnej wycieczce [8].

Komentarz dydaktyczny: W szkole ponadpodstawowej powyższy scenariusz można rozszerzyć o następujące elementy: zapis algorytmu w postaci listy kroków, sprawdzanie, czy dla danego zadania powstały układ par spełnia kryterium trwałości, określenie na czym polegałaby strategia zachłanna w tym algorytmie.

6. Podsumowanie

Metoda Computer Science Unplugged pokazuje, że nauczanie informatyki może odbywać się w sposób angażujący, praktyczny i dostępny dla uczniów na różnych etapach edukacyjnych. Zajęcia realizowane zgodnie z ideą CSU wspierają rozwój myślenia algorytmicznego, umiejętności rozwiązywania problemów oraz współpracy, jednocześnie pomagając przełamywać przekonanie, że programowanie i algorytmika są zagadnieniami trudnymi i przeznaczonymi wyłącznie dla pasjonatów.

Wyniki współczesnych badań dotyczących edukacji informatycznej wskazują, że aktywności realizowane zgodnie z ideą Computer Science Unplugged skutecznie wspierają rozwój myślenia komputacyjnego oraz umiejętności rozwiązywania problemów [2, 6]. Badania porównawcze pokazują również, że uczniowie uczestniczący w zajęciach bez komputera często osiągają równie dobre, a niekiedy nawet lepsze efekty w zakresie rozwijania kompetencji informatycznych niż grupy pracujące wyłącznie przy komputerze, szczególnie na wcześniejszych etapach edukacji [6]. W literaturze podkreśla się także pozytywny wpływ metod unplugged na motywację uczniów, wzrost zainteresowania informatyką oraz budowanie większej pewności siebie podczas nauki programowania [6].

Podejście zgodne z ideą CSU może stanowić wartościowe uzupełnienie tradycyjnych metod dydaktycznych oraz etap przygotowujący do nauki programowania. Dzięki odwołaniu do codziennych doświadczeń uczniów oraz aktywizujących form pracy możliwe staje się przedstawianie nawet złożonych zagadnień informatycznych w sposób bardziej przystępny i zrozumiały.

Wszystkich zainteresowanych praktycznym wykorzystaniem idei Computer Science Unplugged serdecznie zapraszamy na warsztaty.

Literatura

1. Bell T., Witten I. H., Fellows M., Computer Science Unplugged. O informatyce bez komputera... Scenariusze zajęć z dziećmi, tłum. P. Perekietka, Ł. Nitschke, 2008.
2. Chen P., Yang D., Metwally A. H. S., Lavonen J., Wang X., Fostering Computational Thinking through Unplugged Activities: A Systematic Literature Review and Meta-Analysis, *International Journal of STEM Education*, 2023, <https://share.google/sRRUYvj18OYxZ1viR>, ostatni dostęp 30.05.2026 roku.
3. Gale D., Shapley L. S., College Admissions and the Stability of Marriage. Technical report no.1, 1960-61. Prepared under Contract Nonr-562(15) for The Logistic Branch of the Office of Naval Research, <https://web.archive.org/web/20200122083518/https://apps.dtic.mil/dtic/tr/fulltext/u2/251958.pdf>, ostatni dostęp 30.05.2026 roku.
4. Mazur J., Perekietka P., Talaga Z., Wierzbicki J. S., Informatyka na czasie 3. Podręcznik dla liceum ogólnokształcącego i technikum, zakres podstawowy, Wydawnictwo Nowa Era, Warszawa 2021.
5. Ministerstwo Edukacji Narodowej, Podstawa programowa kształcenia ogólnego dla szkoły podstawowej, Warszawa 2017 oraz Podstawa programowa kształcenia ogólnego dla czteroletniego liceum ogólnokształcącego i pięcioletniego technikum, Warszawa 2018.
6. Sun L., Liu J., Liu Y., Comparative Experiment of the Effects of Unplugged and Plugged-In Programming on Computational Thinking in Primary School Students: A Perspective of Multiple Influential Factors, Thinking Skills and Creativity, 2024, <https://doi.org/10.1016/j.tsc.2024.101542>, ostatni dostęp 30.05.2026 roku.
7. Sysło M. M., Algorytmy, WSiP, Warszawa 1997.
8. Sysło M. M., Piramidy, szyszki i inne konstrukcje algorytmiczne, WSiP, Warszawa 1998.
9. Varian H. R., Mikroekonomia. Kurs średni – ujęcie nowoczesne, PWN, Warszawa 2013.
10. Zintegrowana Platforma Edukacyjna, Złożoność obliczeniowa i pamięciowa algorytmu, <https://zpe.gov.pl/a/zlozonosc-obliczeniowa-i-pamieciowa-algoritmu/DF1wJzB8S>, ostatni dostęp 28.05.2026 roku.